An increasing number of studies suggest that the results obtained by computational models are dependent on the interaction topology used within them. That is, in case of agent-based models, the outcome could be much different depending on the number and identity of the agents' partners. In the presence of this observation, the correct experimental methodology would be to treat topology as a parameter. That is, to perform tests with as many relevant topologies as possible, and to assess the results' sensitivity towards them, prior to publication.

Nonetheless, most of today's models operate on a fixed, and often arbitrary topology. This is, in part, due to the ancestry of agent-based simulation. One of the most prominent forerunners of the methodology was cellular automaton. Hence the two canonical ABM topologies: the grid and the torus. While at first these seem as appropriate representations of space, it is obvious that they introduce artificial limitations (e.g., on the number of immediate neighbors, etc.). Furthermore, these topologies are often used to implement abstract concepts of proximity, where their suitability is less obvious.

Recent interest in social networks directed attention towards more general interaction topologies, such as random graphs, "small world" networks, etc. Still, even reports on network-based models rarely account for the sensitivity towards the network's properties. Ultimately, this is due to the canonical way (c.f. "design patterns") agent-based models are built. According to this, the topology is referenced throughout the model, affecting both the agents and the core (model/swarm) class. Therefore, experimenting with different topologies would, in practice, mean to re-implement the simulation a number of times, with all the hassles and risks of such an endeavor.

In this presentation two solutions are offered to this problem. First, RePast's space library is outlined, demonstrating its support for "swappable" spaces. While this approach works well for two-dimensional, grid-like topologies, it
cannot properly handle generalizations to higher dimensions, nor can it bridge the gap between grid-like and network-based topologies.

The second solution overcomes these latter problems, too, by defining the interaction topology in terms of relations between agents. (For example, an agent can acquire a list of all agents that are in the "Neighbor" relation with it.) Naturally, relations can change over time, thus making dynamic topologies possible. Moreover, in this framework, relations are defined and updated by so-called "contexts". The intent is that contexts encapsulate the handling of interdependent relations, making the approach modular and allowing for easy and flexible experimentation with interaction topologies.